Le Probleme de la mise en correspondance : l'etat de l'art

Zhengyou Zhang

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Le problème de la mise en correspondance : L'état de l'art

Zhengyou Zhang

N° 2146

Décembre 1993

THÈME 3

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Le problème de la mise en correspondance : L'état de l'art

Zhengyou Zhang

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Robotvis

Rapport de recherche n° 2146 — Décembre 1993 — 44 pages

Résumé : Le problème de la mise en correspondance est l'un des problèmes les plus difficiles en vision par ordinateur. Nous identifions trois catégories de mise en correspondance : stéréovision, reconnaissance d'objets, et analyse de séquences d'images. Ce rapport vise à faire une revue complète sur l'ensemble de travaux dans la littérature avec une attention particulière sur la mise en correspondance entre deux images au sein d'une séquence, bidimensionnelles ou tridimensionnelles.

Mots-clé : Mise en correspondance, Stéréovision, Reconnaissance d'objets, Séquences d'images

(Abstract: pto)

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

The Matching Problem: The State of the Art

Abstract: The problem of matching is one of the bottlenecks in computer vision. We identify three categories of matching: stereovision, object recognition, and image sequence analysis. This report tries to provide a complet survey of the works reported in the literature, with emphasis on matching between two (2D or 3D) images in a sequence.

Key-words: Matching, Stereovision, Object Recognition, Image Sequence Analysis

Table des matières

Avant-propos

J'ai commencé à travailler sur cet article il y a plus de trois ans. Il y a une abondante littérature sur le problème de mise en correspondance, mais j'ai constaté qu'il n'existait aucun article faisant le point sur l'état de l'art. Mon ambition était alors de faire une revue complète. Je dois admettre que le but est loin d'être atteint et ne le sera pas dans un futur proche à cause de mon emploi du temps. Au lieu de traîner encore longtemps la version incomplète dans un coin de mon tiroir, je pense qu'il est préférable de la publier comme rapport de recherche en espérant qu'elle sera utile pour d'autre chercheurs. Elle leur permettra au moins accès plus facile à la littérature. Tout commentaire ou suggestion est bien venu.

Notons qu'un article par Brown [14] vient d'être publié et que le présent article peut être considéré comme un supplément de celui-ci.

1 Introduction

Le problème de la mise en correspondance, comme celui de segmentation, est l'un des problèmes les plus difficiles, et bien sûr loin d'être résolu, en vision par ordinateur et en vision pour la robotique. Dans un sens large, par correspondance nous entendons l'identification de différents attributs soumis à une certaine relation. Etant donné, par exemple, des points de contours dans une image, pour qu'on puisse obtenir une description polygonale, on doit identifier des points collinéaires constituant un segment de droite, ou bien étendre un segment de droite si d'autres points sont sur la droite. Ici donc, il y a le problème de correspondance entre des points et des segments de droite. Dans ce papier, nous allons passer en revue un autre type de correspondance, où nos données sont des attributs (primitives) dans des images *différentes* et nous devons identifier si celles-ci proviennent provenant d'une *même* primitive. Ce problème est souvent appelé *mise en correspondance* ou problème d'*appariement*.

L'importance de la mise en correspondece ne peut être sous-estimée, même s'il y a des efforts visant à contourner ce problème pour certaines applications. Ses applications incluent vision stéréoscopique, navigation visuelle d'un véhicule mobile, segmentation des scènes complexes, construction de modèles de l'environnement, surveillance dynamique et suivi de cibles.

Selon la nature de l'acquisition d'images, nous pouvons classer le problème de correspondance en trois groupes: appariement stéréoscopique, reconnaissance d'objets et analyse de séquence d'images.

2 Appariement stéréoscopique

La stéréoscopie est une méthode importante pour extraire la structure 3D d'une scène. Elle consiste à déduire le relief à partir de la différence entre des images (souvent 2 ou 3) prises de plusieurs points de vue différents mais en même temps (ce qui n'est pas necessaire si la scène est statique). La géométrie relative des caméras est connue à l'avance. Une revue récente sur des méthodes de la stéréoscopie a été déjà faite par Dhond et Aggarwal [24]. Nous récapitulons ici quelques idées importantes.

Le processus de la stéréoscopie est souvent divisé en trois étapes:

prétraitement: l'extraction de points d'intérêt satisfaisant certaines caractéristiques bien définies.

appariement: la mise en correspondance des points d'intérêt.

reconstruction: l'estimation de la structure 3D de la scène observée.

Les méthodes proposées dans la littérature se distinguent d'abord selon les primitives utilisées: régions ou primitives (points, segments de droite, ou courbes). Elles se distinguent aussi selon la géométrie image utilisée (Les axes optiques sont-ils parallèles ou non? Combien de caméras utilisées, deux ou trois?).

La recherche d'appariements est souvent effectuée en examinant la consistance locale, puis la consistance globale. Après avoir appliqué les contraintes locales, il peut exister des appriements multiples. Pour lever des ambiguïtés, on doit examiner la consistance globale. Les contraintes les plus importantes sont:

1. la contrainte épipolaire,
2. l'unicité de l'appariement,
3. la continuité surfacielle de la disparité sauf sur le bord d'une surface,
4. la continuité figurale de la disparité le long de contours figuraux,
5. la limitation du gradient de la disparité,
6. l'ordre de-gauche-à-droite.

3 Reconnaissance d'objets

3.1 Enoncé du problème

Le problème de la reconnaissance d'objets peut être défini comme suit:

> Nous nous donnons une banque de données où sont stockés des modèles d'objets, et également une vue du monde réel. Pour chaque objet dans la

banque de modèles, le problème de la reconnaissance d'objets consiste à répondre aux questions suivantes:

- L'objet est-il présent dans la scène observée?
- Si oui, quels sont les paramètres de la transformation entre le repère associé à l'objet et celui du capteur?

3.2 Approches principales

La reconnaissance d'objets est un domaine de recherche très actif en vision par ordinateur. Nous listons seulement par la suite des méthodes courantes, et renvoyons le lecteur intéressé à des papiers excellents sur ce sujet [8, 18]. Nous pouvons diviser prèsque toutes les méthodes courantes en les paradigmes suivants:

1. **Recherche arborescente**, par exemple, la méthode de l'arbre d'interprétation contrainte proposée par Grimson and Lozano-Pérez [32, 33]. Etant donné m pri-

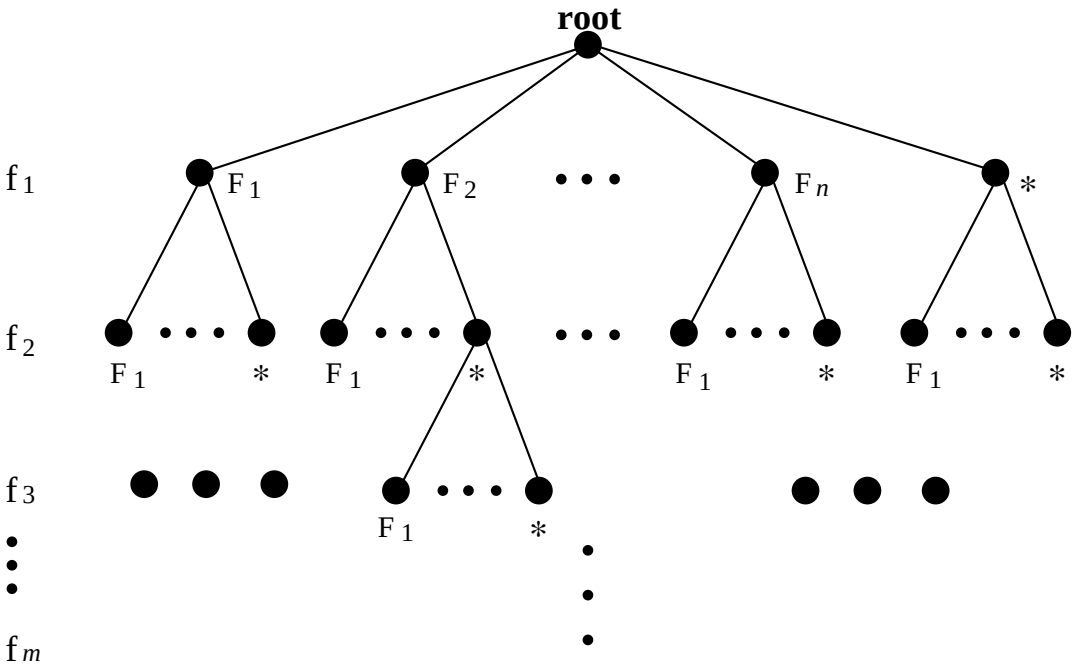

FIG. 1 – *L'arbre d'interprétation pour la mise en correspondance*

mitives $\{f_i,\ i = 1, \ldots, m\}$ dans le modèle et n primitives $\{F_j,\ j = 1, \ldots, n\}$ dans la scène observée. Comme une primitive du modèle peut n'être appariée à aucune primitive dans la scène observée, nous introduisons une primitive spéciale appelée un *caractère nul* (notée par $*$) dans la scène observée. Une liaison entre une primitive du modèle et le caractère nul signifie que la primitive ne peut être appariée à aucune primitive réelle dans la scène observée. Cet appariement ne doit pas être pris en compte dans l'interprétation finale. Maintenant, l'arbre d'interprétation peut être construit comme suit.

Supposons que les primitives du modèle sont ordonnées par une méthode quelconque. Au premier niveau de l'arbre, chaque noeud représente une assignation plausible d'une primitive dans la scène observée à la première primitive f_1 du modèle (voir Fig. 1). Toutes les primitives dans la scène observée, y compris le caractère nul, sont des assignations plausibles à f_1. Donc, au premier niveau de l'arbre d'interprétation, nous avons $n+1$ noeuds. Etant donné chaque assignation hypothétique, nous considérons au deuxième niveau toutes les assignations possibles des primitives dans la scène observée à la deuxième primitive f_2 du modèle. Nous avons donc encore $n + 1$ noeuds sous chaque noeud noeud du premier niveau de l'arbre. Nous pouvons continuer de la même manière jusqu'au moment où toutes les primitives du modèle sont considérées. L'arbre d'interprétation complet a donc m niveaux, chacun correspondant à une primitive du modèle (voir Fig. 1). Une simple énumération montre qu'il y a $(n + 1)^m$ noeuds au niveau m de cet arbre.

Au lieu d'explorer toutes les interprétations possibles des données observées, seulement celles faisables sont retenues en testant la consistance des assignations de primitives du modèle à celles des données. Par consistance nous entendons l'assignation de primitives des données à celles du modèle satisfaisant la contrainte de rigidité. Toute interprétation qui est localement inconsistante est rejetée, ce qui réduit rapidement le nombre de candidats.

2. **Technique d'alignement**. Un alignement est une transformation du repère du modèle en le repère des données. Des travaux représentatifs sont ceux de Ayache et Faugeras [2] (le système `HYPER`), Faugeras et Hébert [26], Huttenlocher et Ullman [43], and Lowe [58, 59] (le système `SCERPO`). Cette technique consiste en deux étapes:

   (a) Calculer des alignements possibles en utilisant un nombre minimum de correspondences entre les primitives du modèle et celles des données. Par exemple, pour m points 3D dans le modèle and n points (2D ou 3D) dans les données, il y a $\binom{m}{3}\binom{n}{3}3!$ alignements possibles, obtenus en assignant chaque triplet de

points du modèle à chaque triplet de points des données. Trois correspondances de points est le nombre minimum pour déterminer une transformation unique. Le groupement perceptuel [58] ou bien la contrainte de rigidité peuvent être utilisés pour réduire le nombre d'alignements hypothétiques.

(b) Vérifier chacun des alignements hypothétiques en transformant les points du modèle dans le repère des données et les comparant avec des données observées. Le processus de vérification consiste à examiner si les primitives du modèle transformées par l'alignement hypothétique coïncident avec des points observés. Cette étape est d'importance cruciale parce qu'on doit éliminer les alignements incorrects mais ne pas perdre les bons.

L'algorithme `RANSAC` proposé par Fischler et Bolles [11] peut être considéré comme une variante de cette méthode. Il choisit d'une manière aléatoire des échantillons d'appariements de primitives du modèle et des données, qui sont utilisés pour calculer une transformation hypothétique. Cette transformation est immédiatement evaluée sur le reste des primitives. Si elle donne un consensus satisfaisant, l'algorithme termine; sinon, l'échantionnage continue jusqu'à trouver une transformation satisfaisante, ou bien il n'arrive pas à détecter l'objet du modèle.

3. **Extension des primitives focales** (**Local-Feature-Focus (LFF) method**). Des travaux représentatifs sont ceux de Bolles et Cain [12] (reconnaissance d'un objet 2D à partir d'une image) et de Bolles et Horaud [39, 13] (reconnaissance d'un objet 3D à partir des données de profondeur). L'idée de base de la méthode LFF est de trouver des primitives proéminentes comme trous, coins, et segments longs. Nous les appelons des *primitives focales*. La sélection d'une primitive focale est basée sur une fonction de plusieurs facteurs, y compris l'unicité de cette primitive, sa contribution espérée, le coût pour la détecter et la vraissemblance de la détection. Chaque primitive focale est utilisée pour prédire quelques primitives voisines pour la poursuite de la recherche. Après avoir trouvé quelques primitives voisines, le programme construit un graphe des primitives du modèle et des données mutuellement consistantes. Une technique de l'appariement de graphes est utilisée pour identifier une clique de primitives des données de taille maximale qui s'apparie avec une clique de primitives du modèle. La clique supérieure à un seuil minimal est considérée comme le meilleur appariement. Bien que l'algorithme pour trouver une clique maximale ait une complexité exponentielle en le nombre de primitives, le graphe est petit puisque la liste des primitives possibles a été réduite à celles près d'une primitive focale, et donc le graphe peut être analysé rapidement.

4. **Hachage géométrique**. Des travaux représentatifs sont ceux de Lamdan et Wolfson [54]. L'idée de base est de précompiler les informations du modèle dans un tableau de hachage avec une représentation appropriée. Considérons, par exemple, la reconnaissance d'objets 2D représentés par des points qui sont soumis à une *similitude* (rotation, translation, plus une échelle). Le hachage géométrique utilise le fait qu'apparier une paire de points du modèle et une paire de points des données définit d'une manière unique la similitude, et le fait que les points préservent leurs coordonnées sous une similitude quelconque s'ils sont représentés dans un repère local attaché aux deux *même* points. Il est divisé en deux procédures: off-line et on-line.
   (a) Dans l'étape du prétraitement (off-line), chaque paire de points ordonnés est utilisée pour définir un repère orthogonal, et la paire est appelée une *paire de base*. Tous les autres points du modèle sont représentés par des coordonnées dans ce repère. Chaque coordonnée (après une discrétisation appropriée) est utilisée comme une entrée au tableau du hashage, qui stocke toutes les paires (*modèle*, *paire de base*) pour chaque coordonnée.
   (b) Dans l'étape de la reconnaissance (on-line), une paire de points ordonnés arbitrairement dans une image est choisie et les coordonnées des autres points sont calculées dans cette base. Pour chaque coordonnée, examiner une entrée appropriée dans le tableau du hachage et voter pour toutes les paires (*modèle*, *paire de base*) qui sont là. Si une certaine paire (*modèle*, *paire de base*) obtient le meilleur score, elle est considérée correspondre à celle choisie dans la scène. La transformation de similarité entre les repères de coordonnées correspondant à ces paires de base peut être calculée, et peut éventuellement être recalculée en utilisant des appariements supplémentaires.
5. **Transformation de Hough généralisée** ou **Clustering de poses**. La transformation de Hough a été introduite pour détecter des primitives géométriques simples comme des droites dans une image en accumulant des évidences dans leur espace paramétrique. Elle a été généralisée pour reconnaitre et localiser des objects en considérant l'espace de transformation [23, 74]. Cette méthode peut être divisée en deux étapes:
   (a) La première étape est de générer des candidats de transformations/poses et les découper dans l'espace de transformation. Pour calculer une transformation/pose unique, un nombre minimum, disons k, de primitives dans un modèle doivent être appariés avec le même nombre de primitives dans les données. Par exemple, pour reconnaitre des objets 2D représentés par

des points à partir des données 2D représentées aussi par des points, deux appraiements sont nécessaires, i.e., $k = 2$. Pour chaque appariement de k primitives entre le modèle et les données, essayer de calculer une transformation/pose. Si elle existe, découper-la dans l'espace de transformation (après une discrétisation appropriée), qui peut être représentée par un tableau de l'accumulateur. Chaque élément du tableau enregistre l'accumulation des évidences pour une transformation/pose donnée (plus strictement, un ensemble de transformations/poses dans l'intervalle de la quantisation).

(b) La deuxième étape est d'effectuer le groupement dans l'espace de transformation, par exemple en balayant le tableau de l'accumulateur, pour obtenir la(les) meilleure(s) transformation(s)/pose(s) et son(leurs) support(s). Dans [74], Stockman propose un algorithme de clustering, basé sur la distance entre des candidats de transformation/pose, pour détecter la transformation/pose avec le support le plus important. Cette approche évite le problème à cause de la discrétisation de l'espace de transformation.

Le lecteur peut trouver beaucoup de similarités entre cette méthode et la dernière. La différence principale est l'espace de hashage utilisé. Cette méthode utilise l'espace de transformation/pose, quant à la méthode de hachage géométrique, elle utilise l'espace géométrique (des coordonnées relatives à un repère choisi).

6. **Recherche basée sur la CAO** (conception assistée par l'ordinateur). Des travaux représentatifs sont ceux de Hansen et Henderson [34], et Ikeuchi et Kanade [44, 45]. Les méthodes dans cette catégorie exploitent la connaissance géométrique dans des modèles CAO pour générer, d'une manière automatique, des stratégies de la reconnaissance. Un objet dans la banque de modèles est souvent représenté par un *graphe d'aspect*. *L'aspect* est l'apparence topologique d'un objet d'un point de vue particulier. Un petit changement de point de vue change la taille des primitives (contours et surfaces), mais ne provoque pas leur apparition ou disparition. Quand un petit changement de point de vue provoque l'apparition ou la disparition d'une primitive, un *événement* se produit. Un graphe d'aspect est construit en représentant des aspects comme noeuds et des événements comme un chemin entre les noeuds correspondants. En analysant la robustesse, l'état complet, l'unicité et le coût de détection des primitives dans les aspects, un arbre de stratégies est construit pour chaque objet. Il décrit, d'une manière systématique, le processus de recherche utilisé pour la reconnaissance et la localisation d'un objet particulier dans une scène donnée. Durant la phase de reconnaissance, les arbres de stratégies guident la recherche et donc la vitesse de reconnaissance s'accroit.

7. **Raisonnement basé sur l'évidence**. Jain et Hoffman [46] ont développé un algorithme basé sur le raisonnement de l'évidence pour reconnaître des objets 3D à partir d'images de profondeur. L'idée de cette méthode est: plutôt que d'utiliser toutes les informations fournies par une représentation, il est possible d'utiliser seulement des informations ou des évidences proéminentes, qui donnent la réplique à certains objets. Une collection d'évidences peut être utilisée pour déterminer le contenu vraisemblable d'une scène observée. Le système de reconnaissance décrit dans [46] est implémenté comme un système à règles de production. La partie prémisse est composée des bornes de paramètres caractéristiques d'une facette de surface comme le périmètre, la superficie et le sens de la surface (planaire, convexe, ou concave), et la relation entre les voisins adjacents ou de saut comme l'angle entre les normaux et la distance entre deux facettes. La partie action d'une règle est composée des poids de pondération des évidence pour chaque objet dans la banque de modèles si la condition est satisfaite. Après avoir activé toutes les règles, un raisonnement basé sur les évidences est appliqué pour décider si un objet est observé dans la scène ou non. La partie essentielle et difficile de cette approche est de capturer les propriétés d'un objet les plus générales possibles, d'un côté, pour qu'on ne manque aucun objet à cause du bruit ou de la distortion introduit durant l'acquisition, et les plus discriminatoires possibles, d'un autre côté, pour qu'on ne confonde pas d'objets différents. Cette approche a été testée surtout avec des scènes contenant un seul objet.

Une caractéristique partagée par les méthodes de reconnaissance d'objets est une procédure de prétraitement, qui compile des informations dans les modèles pour simplifier le processus de reconnaissance. Cette procédure est exécutée *off-line*, et est souvent très coûteuse si l'on utilise au maximum les informations contenues dans les modèles.

4 Analyse de séquence d'images

4.1 Enoncé du problème

Le problème de correspondance dans une séquence d'images est beaucoup plus dure que dans un système stéréoscopique. Une raison principale est la manque de connaissance du mouvement d'un objet d'une image à l'autre. Dans la stéréo, la relation géométrique entre les caméras est connue, qui est traduite en la contrainte épipolaire. Cette contrainte permet de réduire la région de recherche de la correspondance d'une primitive de deux dimensions à une dimension. Dans une séquence d'images,

puisqu'on connaît pas le mouvement, on doit, en principe, chercher la correspondance dans toute image. Un autre facteur qui complique le problème est le problème d'objets multiples. C'est souvent beaucoup plus simple de mettre en correspondance un groupe d'objets soumis à un seul mouvement que des groupes d'objets soumis à des mouvements différents. Si des primitives sont soumise à un seul mouvement, nous pouvons utiliser une cohérence entre les correspondances de primitives, mais avec des mouvements différents, cette cohérence est perturbée. D'autres facteurs qui rendent le problème difficile sont:

- **Occlusion:** Un objet mobile peut être caché partiellement ou totalement par d'autres objets.
- **Disparition:** Un objet mobile visible dans le champ de vue à l'instant peut le quitter partiellement ou totalement dans les instants suivants.
- **Apparition:** Un objet invisible auparavant peut entrer partiellement ou totalement dans le champ de vue.
- **Absence:** Avec l'état de l'art des méthodes pour la détection de primitives, nous rencontrons souvent le problème que des primitives qui auraient dû être présentes mais ne le soient pas à cause de l'échec du processus de l'extraction (ou reconstruction) de primitives.

Nous pouvons classer les méthodes proposées dans la litérature en deux approches générales selon le niveau de l'abstraction des informations dans des images: l'approche iconique ou l'approche basée sur des primitives.

4.2 Approche iconique

La première est directement basée sur la représentation iconique/pixellique. Elle est souvent connue sous le nom de *template matching* ou de la mise en correspondance par la corrélation [3, 31].

Un *template* ou une zone d'intérêt, qui est en fait une sous-image de la première image, est d'abord choisi. Plus formellement, cette zone est décrite par

$$T_1(\mathbf{x}) = I_1(\mathbf{x}) \ , \quad \mathbf{x}_1 \leq \mathbf{x} \leq \mathbf{x}_2 \ , \tag{1}$$

where $\mathbf{x}$ est le vecteur de coordonnées image et $I_1(\mathbf{x})$ ($[0,0]^T \leq \mathbf{x} \leq [M-1, N-1]^T$) est la première image avec dimension $M \times N$. Supposons que la deuxième image est $I_2(\mathbf{x})$. Pour trouver la correspondance de $T_1(\mathbf{x})$ dans $I_2(\mathbf{x})$, nous devons calculer une

mesure de similarité entre $T_1(\mathbf{x})$ et une partie de $I_2(\mathbf{x})$ (i.e., une sous-image de $I_2(\mathbf{x})$ ayant la même forme et la même dimension que $T_1(\mathbf{x})$). Une mesure standard est la distance Euclidienne $d(\mathbf{y})$ au carré, donnée par

$$d^2(\mathbf{y}) = \sum_{\mathbf{x}=\mathbf{0}}^{\mathbf{x}_2-\mathbf{x}_1} [I_1(\mathbf{x}+\mathbf{x}_1) - I_2(\mathbf{x}+\mathbf{y})]^2 \ , \tag{2}$$

où la sommation est effectuée sur toute la zone de *template*. Si la deuxième image sur point $\mathbf{y}$ est une correspondance exacte de $T_1(\mathbf{x})$, alors $d(\mathbf{y}) = 0$; sinon $d(\mathbf{y}) > 0$. En pratique, la partie de $I_2(\mathbf{x})$ minimisant $d^2(\mathbf{y})$ est considée comme la correspondance de $T_1(\mathbf{x})$.

Une autre mesure qui est couramment utlisée s'appèlle la *corrélation croisée normalisée*:

$$C(\mathbf{y}) = \frac{\sum_{\mathbf{x}=\mathbf{0}}^{\mathbf{x}_2-\mathbf{x}_1} I_1(\mathbf{x}+\mathbf{x}_1) I_2(\mathbf{x}+\mathbf{y})}{\left(\sum_{\mathbf{x}=\mathbf{0}}^{\mathbf{x}_2-\mathbf{x}_1} I_1^2(\mathbf{x}+\mathbf{x}_1) \sum_{\mathbf{x}=\mathbf{0}}^{\mathbf{x}_2-\mathbf{x}_1} I_2^2(\mathbf{x}+\mathbf{y}) \right)^{1/2}} \ . \tag{3}$$

$C(\mathbf{y})$ sera maximisée si la partie de I_2 au point $\mathbf{y}$ est la correspondance de $T_1(\mathbf{x})$.

L'avantage de cette approche est que l'effort sur le prétraitement d'image est minimal. Mais toutes ces deux mesures sont très sensibles au bruit, au changement d'intensité, au déplacement non-linéaire, et au déplacement dont la direction de translation n'est pas paralléle au plan de l'image.

Récemment, Chou et Chen [19] ont proposé une autre méthode pour le *template matching*, qui réduit considérablement le temps de calcul mais avec un peu moins de performance. Cette méthode consiste à transformer d'abord les images grisées en images binaires, tout en conservant les moments dans une bloque d'image: la moyenne d'échantillonage et l'écart type d'échantillonage. La fonction de corrélation peut ensuite être synthétisée comme le nombre de paires de pixels qui ont la même valeur, qui est calculé facilement avec l'opération logique `AND` et l'addition.

4.3 Approche basée sur des primitives

La deuxième approche utilise des primitives géometriques, ou *tokens* en anglais. Les primitives couramment utilisées sont des points, contours, segments de droite, courbes, plans, ou même des surfaces. Les attributs photogramétiques, comme le gradient d'intensité, peuvent être utilisées pour renforcer les contraintes pendant la mise en correspondance. Avec cette approche, une image peut être décrite comme

un graphe avec primitives comme noeuds. Le problème de la mise en correspondance de deux images devient la correpondance entre deux graphes: *isomorphisme de sous-graphes*. Ce problème a été bien étudié dans la théorie du graphe. Malheureusement, tous les algorithmes ont une complexité NP-complète, et donc en pratique ils ont en général peu d'intérêt. Pour développer des algorithmes rapides, les chercheurs en vision par ordinateur essaient intégrer des informations a priori, par exemple, la contrainte de rigidité (le mouvement est rigide), ou/et des informations physiques, par exemple, l'accélération limitée.

Comme pour l'approche iconique, la mise en correspondance entre des primitives implique une mesure de qualité. La définition d'une telle mesure dépend du problème à la main, mais elle est souvent composée de trois parties. La première intervient la similarité entre les attributs de la paire de primitives en question. La deuxième intervient la similarité entre les relations entre primitives dans une image et les relations entre primitives dans l'autre image. La troisième intervient la pénalité (un nombre négative) pour les primitives non-appariés dans une ou l'autre image. Le processus de la mise en correspondance doit maximiser la mesure de qualité.

La mise en correspondance entre deux images est très difficile comme décrit par la suite. Récemment, certains chercheurs réalisent que le problème devient beaucoup plus simple si l'on utilise une séquence longue d'images prises à un court intervalle de temps. En vérité, comme l'intervalle de temps est petit et la vitesse d'objets est contrainte par les lois physiques, les déplacements d'objets entre des images successives sont limités, c'est-à-dire que la correspondance d'une primitive à l'instant suivant doit être dans son voisinage. De plus, des objets souvent se déplacent de manière lisse [77, 48, 72], la cohérence du mouvement peut être utilisée pour prédire la position des primitives dans le temps à venir, qui réduit considérablement l'espace de recherche de correspondances. Les techniques statistiques d'association des données [4, 9] sont souvent utilisées pour résoudre le problème de mise en correspondance et celui de l'estimation du mouvement [21, 82, 79]. Elles sont déjà reconnues comme une approche efficace. Dans le reste de cet article, nous considérons principalement les méthodes différentes proposées dans la littérature pour mettre en correspondance des primitives dans une séquence d'images.

5 Correspondance entre deux images

5.1 Structures relationnelles

Traditionnellement, le problème de correspondance est posé pour apparier des primitives entre deux images d'une scène dynamique. La différence entre les deux

images peut être assez grande (*long range motion* en anglais). Dans pratiquement tous les algorithme de matching, l'hypothèse du type *rigidité* est imposée sur la configuration de primitives pour obtenir des appariements corrects. Ceci est compatible avec des études psychologiques qui montrent que notre système visuel accepte seulement quelques (souvent une) configurations satisfaisant la contrainte de rigidité: Tout ensemble de primitives soumises à une transformation 2D qui a une interprétation unique comme un objet rigide en mouvement dans l'espace doit être ainsi interprété [76].

Le problème général de la correspondance peut être formulé comme le matching entre des structures relationnelles. Étant donné des primitives distinguées dans une image, elles sont connectées par des relations binaires et même ternaires comme, par exemple, la distance entre deux points, l'angle entre deux droites, un point est sur une droite, trois droites s'intersectent sur un point. Une relation peut être N-aire pour un N arbitraire, mais en pratique $N = 2$ est souvent utilisé, et une structure relationnelle devient un graphe standard. Étant donné deux telles structures, la tâche de la mise en correspondance consiste à identifier une sous-structure *identique* dans les deux structures. Toutefois, une sous-structure *identique* ne peut rarement être trouvée, parce que

- les primitives détectées sont bruitées, mal localisées,
- il y a des distorsions dans les images acquises, et
- un mouvement rigide dans l'espace 3D induit généralement à des déplacements différents de primitives dans l'image.

D'où l'introduction du concept de *matching inexact*: détermination d'une sous-structure *compatible* qui tolère des variations attendues ou non [73]. Il faut noter que le matching de deux structures relationnelles peut être considéré comme le problème de l'étiquetage cohérent par satisfaction de contraintes (relations unitaires ou binaires) [36, 37].

5.2 Relaxation

Une méthode populaire est la technique de relaxation [70, 52, 25, 42]. C'est un algorithme itératif et localement parallèle. D'abord, un ensemble initial des appariements possibles est construit en appariant chaque primitive de la première image avec chaque primitive de la deuxième image. Cet ensemble est organisé comme une collection de nœuds $\{a_i\}$, un nœud pour chaque primitive de la première image. Associés à chaque nœud a_i sont le vecteur $\mathbf{x}$ qui représente l'information géométrique de la primitive dans la première image, et l'ensemble d'étiquettes L_i qui représente les

appariements possibles de la primitives. Une étiquette spéciale dite *le caractère nul* $\star$ est initialement inclus dans L_i, qui signifie que la primitive en considération n'a pas de primitive correspondante dans la deuxième image. A chaque étiquette l dans L_i nous associons un nombre $p_i(l)$ qui est interprété comme une estimée de probabilité que l est effectivement la correspondance de la primitive en considération dans la première image. Donc nous avons $p_i(l) \in [0,1]$ et $\sum_{l \in L_i} p_i(l) = 1$. Ces estimées de probabilité seront au fur à mesure mises à jour en tenant compte de la cohérence dans le voisinage. Si relativement beaucoup de primitives dans le voisinage sont compatibles avec l'appariement l, alors $p_i(l)$ augmentera; sinon, $p_i(l)$ diminuera. Après un nombre suffisant d'itérations, un nœud est considéré comme appariable s'il a une étiquette avec une probabilité suffisamment élevée, et non appariable sinon. Ce qui est essentiel dans cette technique est la définition de la règle pour la mise à jour des probabilités.

Pour réduire les ambiguïté des appariements initiaux, des contraintes unitaires peuvent être imposées: l'angle du coin, la grandeur du gradient, la longueur du segment de droite, le maximum de la disparité[5], etc. Les contraintes binaires utilisées déterminent la manière de l'évolution des probabilités. Barnard et Thompson utilisent dans [5] des points d'intérêt (où la variance est forte dans toute direction) et montrent l'efficacité de la propriété de la continuité local de la disparité. Cette propriété est une conséquence directe de la continuité des surfaces réelles dans l'espace. Leur méthode a donc des difficultés près de la discontinuité. Ogawa [63] utilise aussi une technique de relaxation pour apparier deux ensembles de points. Les points représentent des étoiles, chacune ayant une magnitude qui peut être utilisée comme une contrainte unitaire. Au lieu d'utiliser des points individuellement, il considère des paires de points comme primitives. La mesure de compatibilité'e est calculée en tenant compte de la transformation géométrique à partir des deux paires en considération. Cheng et Huang [17] vont encore plus loin. Les segments de droite 2D sont utilisés. Au lieu d'utiliser des segments individuels, ils utilisent une représentation qu'ils appellent une *structure étoile*. Une étoile est un groupe de segments qui sont parallèles, col-linéaires, ou adjacents. La technique de relaxation est utilisée, mais comme les primitives maintenant sont des étoiles qui contiennent beaucoup plus d'informations, l'espace de recherche est considérablement réduite. La littérature sur la relaxation est abondante. On peut encore citer [20, 49, 67, 55].

5.3 Recherche arborescente

L'approche traditionnelle d'attaquer la mise en correspondance est la recherche arborescente en profondeur d'abord[37, 41], comme celle décrite dans la section 3.2

sur la reconnaissance d'objets. Pour expliquer une telle procédure, nous appelons les primitives dans la première images des unités, et celles dans la deuxième image des étiquettes. La procédure assigne des étiquettes aux unités tant qu'elle peut trouver une étiquette pour chaque unité qui est compatible avec étiquettes déjà fixées à des unités précédentes, compatible au sens des contraintes binaires définies. Aussitôt que la procédure n'arrive pas à trouver une étiquette pour une nouvelle unité, elle rebrousse le chemin (*backtrack*). Il se peut donc avoir beaucoup de rebroussements avec l'algorithme standard. Haralick et Elliott [35] montrent analytiquement et expérimentalement que des heuristiques comme suit donnent des performances supérieures à l'algorithme standard:

- Anticiper la future et essayer d'abord les endroits où on sera très probablement amené à échouer.
- Mémoriser ce qu'on a déjà fait pour éviter de commettre la même erreur.

5.4 Détection de la clique maximale

Une technique bien connue pour l'isomorphisme du sous-graphe est la détection de la clique maximale, qui a été appliquée dans l'analyse de séquence d'images [3, 66]. D'abord, un graphe d'association est construit de la manière suivante. Pour chaque primitive $\mathbf{p}_1$ de la première image et chaque primitive $\mathbf{p}_2$ de la deuxième image, elles forment un nœud a_i si elles satisfont les relations unitaires. Deux nœuds a_i et a_j se connectent s'ils sont compatibles, i.e., s'ils satisfont les contraintes binaires. Le meilleur matching entre deux structures relationnelles est le plus grand ensemble de nœuds qu'ils sont mutuellement connectés dans le graphe d'association — la clique maximale. Des algorithmes pour la détection de la clique maximale existent dans la littérature (voir [66] pour les références), à savoir que la détection de la clique maximale peut être coûteux en temps de calcul. Notons aussi que cette technique a été appliquée aussi pour la reconnaissance d'objets, par exemple, dans [12], où la taille du graphe est bien limitée en utilisant des primitives focales (voir la section 3.2).

5.5 Programmation dynamique

La programmation dynamique est une méthode pour résoudre des problèmes d'optimisation non-linéaire quand les variables ne sont pas toutes interdépendantes. Un tel problème peut être décomposé en une séquence de sous-problèmes d'optimisation avec qu'une variable (plus général, avec un sous-ensemble de variables). Considérons

le problème suivant

$$\min_{x_1,x_2,x_3} h(x_1,x_2,x_3) \ ,$$

où x_i prennent des valeurs discrètes. Si on connaît que x_2 dépend seulement de x_1, et que x_3 dépend seulement de x_2, c-à-d,

$$h(\cdot) = h_1(x_1,x_2) + h_2(x_2,x_3) \ ,$$

on peut d'abord minimiser h_1 sur x_1 et mettre dans un tableau la valeur minimum de $h_1(x_1,x_2)$ pour chaque x_2, i.e.,

$$f_1(x_2) = \min_{x_1} h_1(x_1,x_2) \ .$$

Continuons de la même manière et éliminons x_2 en calculant $f_2(x_3)$ comme

$$f_2(x_3) = \min_{x_2}[f_1(x_2) + h_2(x_2,x_3)] \ .$$

Le problème initiale devient donc

$$\min_{x_1,x_2,x_3} h(x_1,x_2,x_3) = \min_{x_3} f_2(x_3) \ .$$

La programmation dynamique est équivalente à la recherche d'un chemin optimal dan un graphe. Son application à la mise en correspondance est illustrée dans la figure 2. Les primitives de la première image sont ordonnées, de la manière arbitraire ou non, et sont indiquées sur l'axe i. Les primitives de la deuxième image sont indiquées sur l'axe j. (Supposons qu'on a N et M primitives dans les deux images, respectivement.) Un chemin de $i = 1$ à $i = N$ est une association des primitives de la deuxième image à celles de la première image. Le problème de matching est donc de trouver le chemin optimal. Le principe de la séparabilité dans la programmation dynamique est appliqué de la manière suivante. A chaque étape i, un nœud (i,j) est associé au chemin qui garantit que le coût cumule est minimale jusqu'à l'étape i. Le coût cumulé du chemin terminé au nœud (i,j) est défini comme

$$C(i,j) = \min_{1\le k\le M} f[C(i-1,k), c(i-1,k;\ i,j)] \ ,$$

où c est le coût local de lier le nœud $(i-1,k)$ au nœud (i,j), et $f(C,c)$ est une combinaison appropriée quelconque de C et c. La complexité de cet algorithme est $O(NM^2)$. Si les primitives dans les deux images sont ordonnées de la même manière, ou si on a d'autres informations, l'espace de recherche peut être considérablement réduite.